Company/Institution Name

Matched

- Shujaullah Ahsan, shujaullah.ahsan001@umb.edu
- Nel Richard Barthelemy, nel.barthelemy001@umb.edu
- Nakra Ath-Ly nakra.athly001@umb.edu
- Praladh Chaulagain p.chaulagain001@umb.edu
- Khezir Malik khezir.malik001@umb.edu
- Client: Thomas Hüpper, thomas.huepper@gmail.com

Project Proposal v1.0 02/25/2022

The project Proposal looks good. Can you please elaborate purpose section along the lines of issues athletes facing now after their active career and how Matched will be helping them to shape their career and generate income. It will be great if you can breakdown resources and mention specific user activities in Workflow diagram in page 6. Thanks!

Overall the proposal is good. Just a few things can be improved. 1) Font size in the figures on page 7 is too small. The font size should not smaller than the normal paragraph font size. 2) You can refer your figures in the paragraph, so the reviewer could know what's the relation between your texts and figures. 3) section 3.5 need more details, for example, what information should be collected through this page. The detail could either in figure caption or normal paragraph.

The proposal looks good. For the purpose section, please provide detail explanation about the current challenge and your solution. the figures need detail text explanation in section 3, specially for each figures.

```
Technology choices look reasonable and UI sketches look great.
Good job! Score 100/100.
```

1 Introduction

1.1 Purpose

The website aims to help retired athletes who are looking for a new career.

1.2 Scope

The software product name will be **matched**

The list of things software will do:

- Matched will record the interests and athletic background of the user.
- The website will match the users according to their data to the companies for jobs and their mentorships.
- It also matches users to college degree programs according to their interests and skill levels.
- Also allows users to share their success stories and accomplishments so users can see the other people success stories.

The website will not do the following:

- The matched will not have adds.
- The website is only for localized users (only for US people)
- The website will not the share the data of the users to any other platform.

1.3 Overview

This software aims to have a website that allows athletes to make their profile; then, it will record their interests and background and connect them to the companies that can offer them jobs or mentors, or the website also suggests that different college programs start a new career after their retirement.

1.3.1 Existing Work

The current work we have is the registration and login page. Over the winter break, we worked on a Django project to rewrite the CS portal. we can use and modify some of that code to add users to the database.

1.3.2 Limitations

- We might not be able to use ReactJS due to our team's inexperience with the framework.
- At first, we have to create our mock jobs data, and there wouldn't be that many because of using an API.

1.4 Definitions

- HTML - Hyper Text Markup Language
- JSON - JavaScript Object Notation
- SQL LITE- For the data base.
- Django- The python framework that will help in maintaining the database for the website.
- CSS - Describes how HTML elements are to be displayed on screen, paper, or in other media.
- ReactJS- Building user interfaces specifically for single-page applications.
- Python- To build websites and software, automate tasks, and conduct data analysis

2 Specifications

- Django
  - We use Django to store and maintain the user model, employers, jobs model, mentors model
    - The user model will keep track of the first name, last-name, username, date joined, former sport, field of choice, job applications,mentor
    - The jobs table will have a list of available jobs
    - The employer table will have the company name, the job listings for that company, and the user applications
    - mentors table will just mentors and their mentees
- ReactJS
  - JSX , CSS to style pages can create components
  - navigation bar component
    - will be present in most views
  - career dashboard will have views for
  - user info page (the current user's information)
  - employer page (employers can create and post job listings)

3 Design

3.1 Use Cases

The audiences are athletes that are retired or consider getting a new career.
(**Athletes**)

- Mentor ship
- Job
- School
- Success stories

(**Employers**)

- Create job listings
- View potential employees

Cases
Athletes

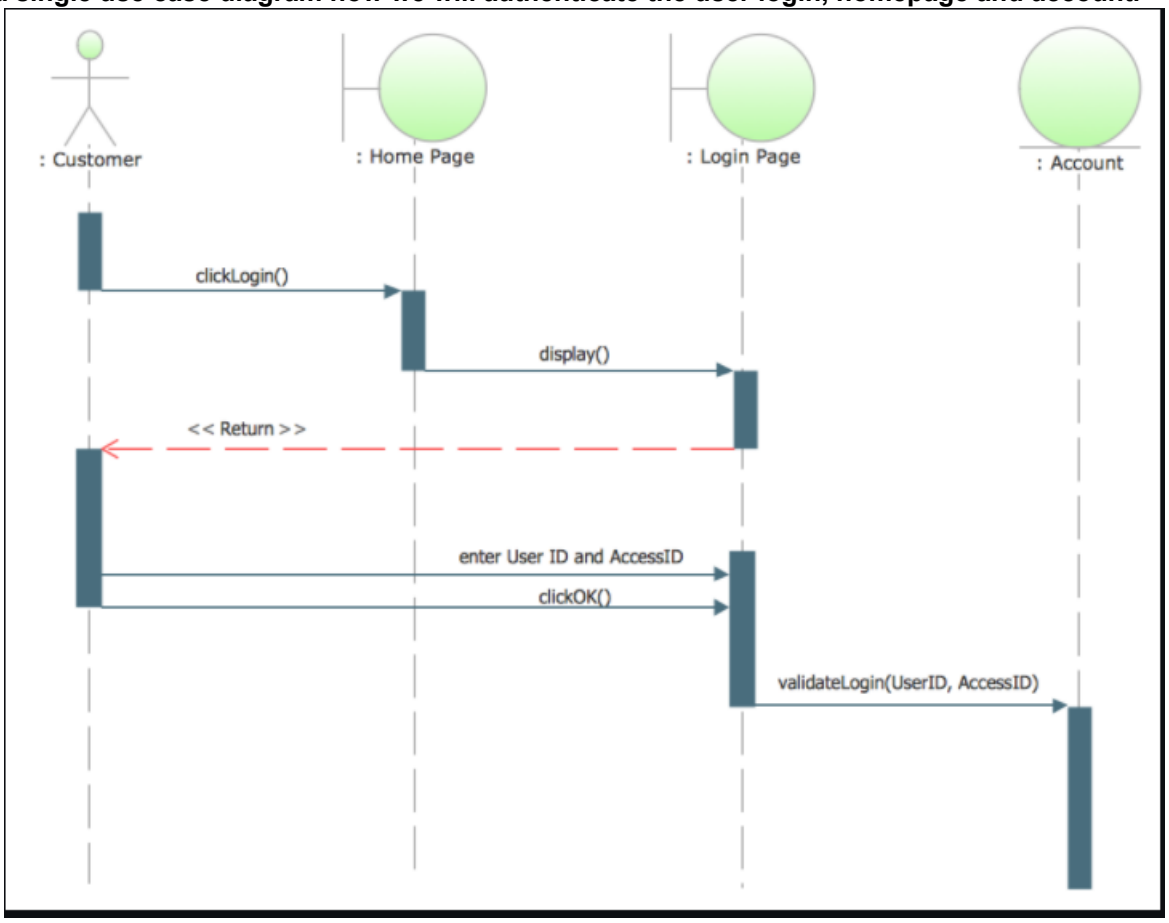
- Mentor ship - It would allow the client to get help to see what they want to do after they retired. They can help them get a job or try to guide their life to the path they want.
- Job - They would submit what career they want to pursue. It would get submitted to all the jobs that they filled in and maybe jobs will hire them.
- School - Some may want to pursue schooling after retiring. They can check what schools in the area or what major they want to attend.
- Success Stories - They would put what they did great as an athlete and their greatest accomplishment. They can add any information they would like to share to people.

Employers

- Create job listing- Employers would fill out a form to make job listing allowing them to hire people.

- View Potential employees - This allows the employer to check all the users profiles to see if they are interested in hiring and mentoring.

There is a single use case diagram how we will authenticate the user login, homepage and account.



Found the diagram: `https://python.astrotech.io/design-patterns/uml/sequence-diagram.html`

3.2 Architecture

We using the Three tier architecture to organize and categorize our development. Our reasoning for this is that required platforms for the project each perform the roles assigned-in a three tear architecture model. User data and employer data is stored in the database, the front end will use api calls to the database to access this data, the data will be displayed using jsx (part of reactJs).

Presentaion Tier: We will have the HTML, CSS and JS for the front end thier on which our UI will be based on. We also use REST API calls to SQL lite data base.

Application Tier: The application tier based on the data base access logic where we will use python and JS to access the data base and Django framework.

Data Tier: We will use Django python framework to acess the and store our data.

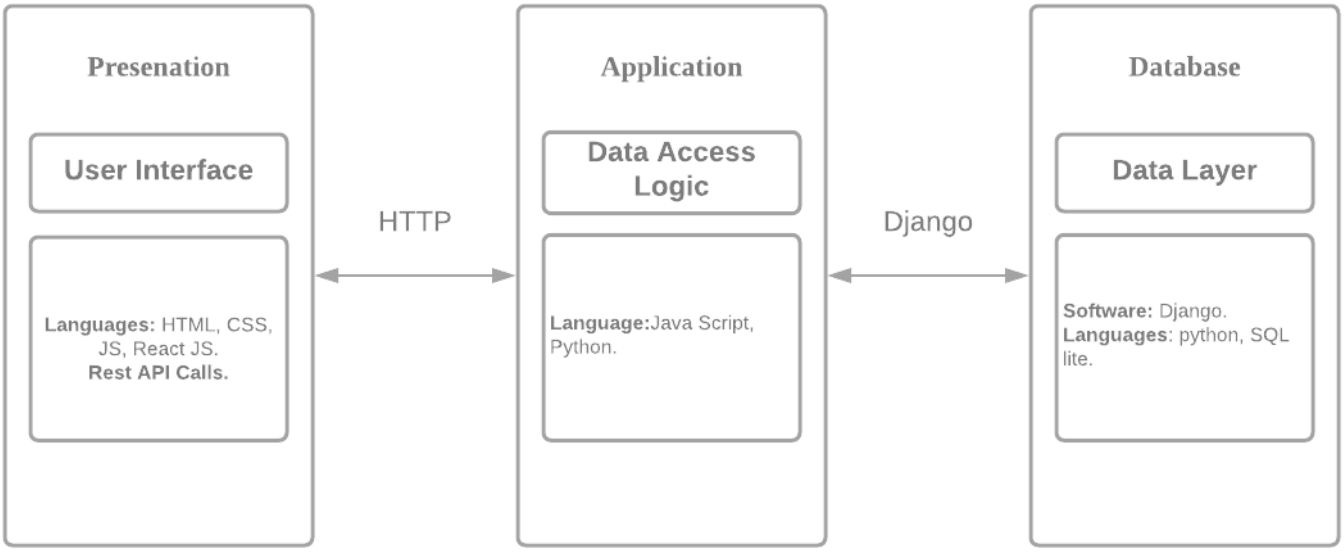


Fig: The Three-Tier Architecture

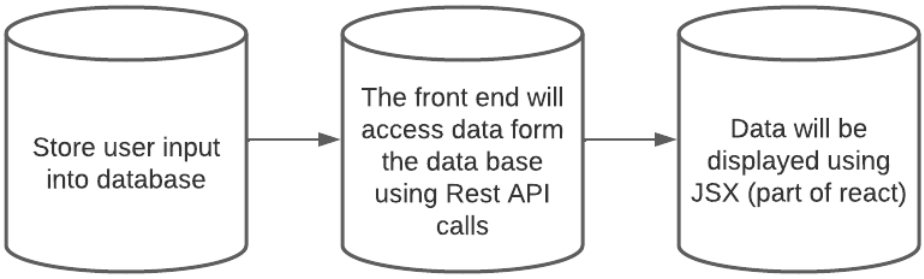


Diagram of how the user's data will be used

3.3 Work Flows

1. When first visiting the Matched website, the user will have the Login or Registration option.
2. If the user is new to Matched, will create an account and then be redirected to the Login page.
3. When the user login successfully then user will directed to Matched dashboard where user will be able to see the available jobs, mentor and success stories of other athletes.
4. The dashboard page also has resources, profile, and logout options.
5. User can access the profile from dashboard so user can see the profile or change the profile info according to the user's needs.
6. The user data will be stored in Django using SQL lite.

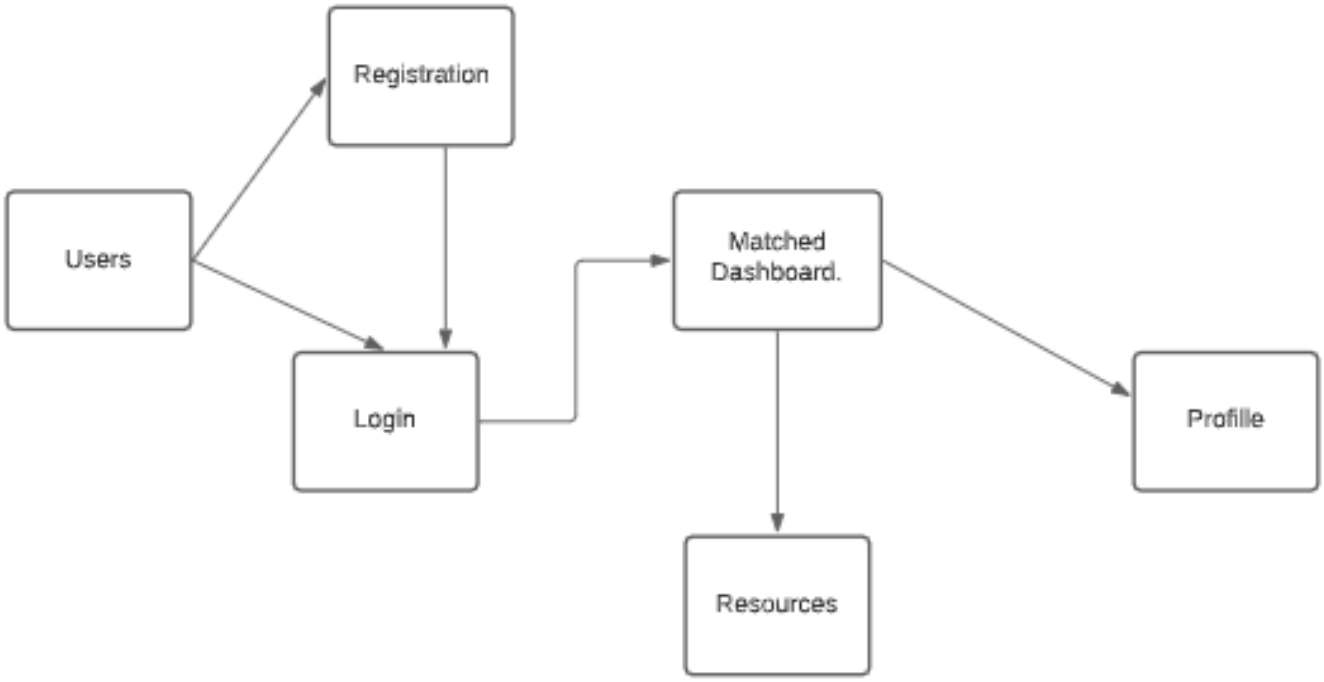


Diagram of showing the workflow.

3.4 Technologies and Implementation Details

ReactJS and Django

- we will use ReactJS for the front end
  - ReactJS uses JSX
    - JSX is a mix of HTML and JavaScript
    - We will use CSS and Bootstrap
      - CSS allows for styling HTML elements
      - Bootstrap comes with predefined styles that will speed up the styling process
  - We will use Django
    - Django allows us to connect the front end with the database
    - Using SQLlite, we can create database tables to store user data
    - We can also create a REST API using Django, then we can make API calls form the front end to access the data directly

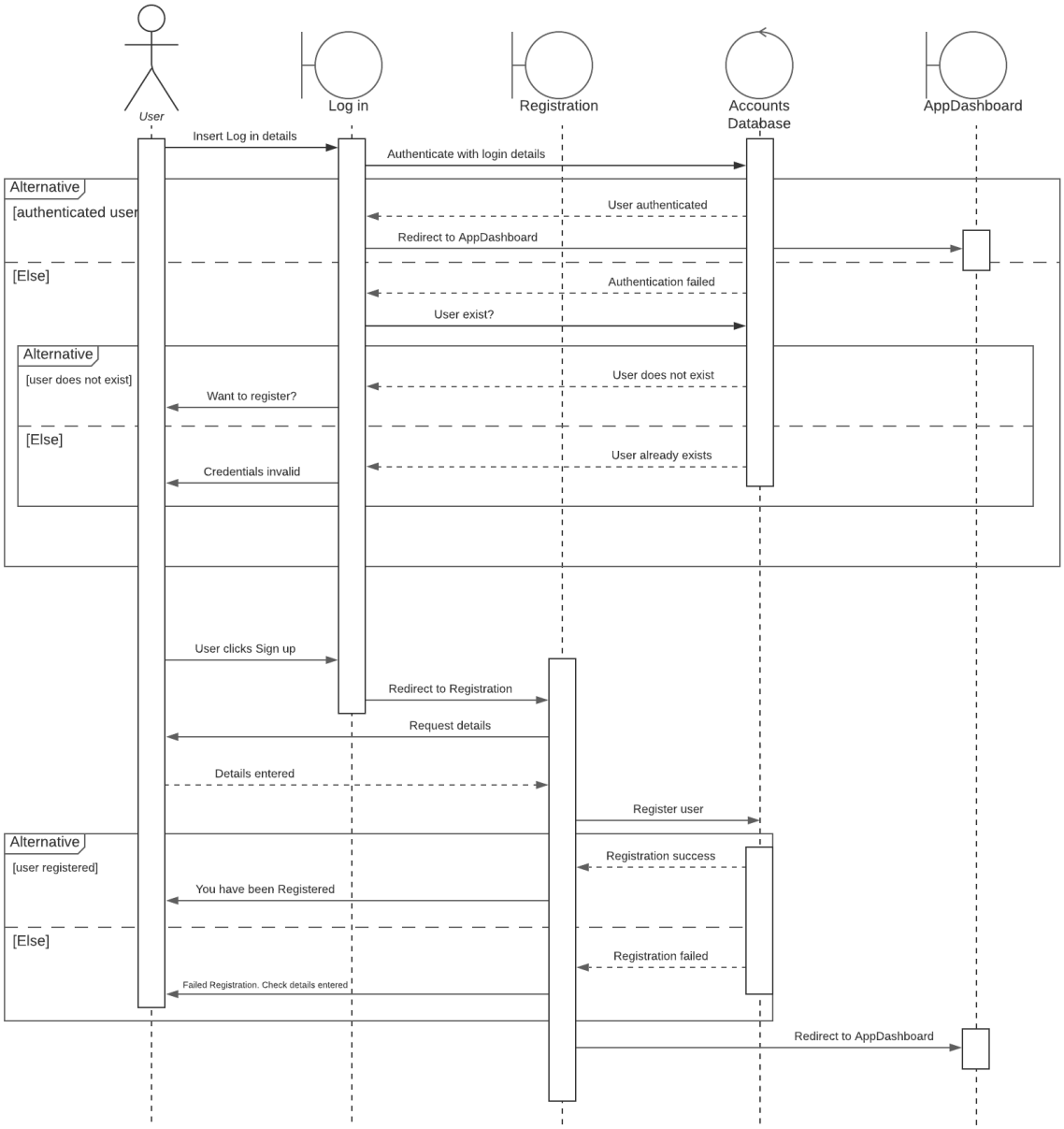


Fig: Sequence diagram of how the Matched work.
`https://stackoverflow.com/questions/66265720/sequence-diagram-including-registration-and-login`

3.5 User Interface

We used the Figma to generate the UI interface for the Matched.

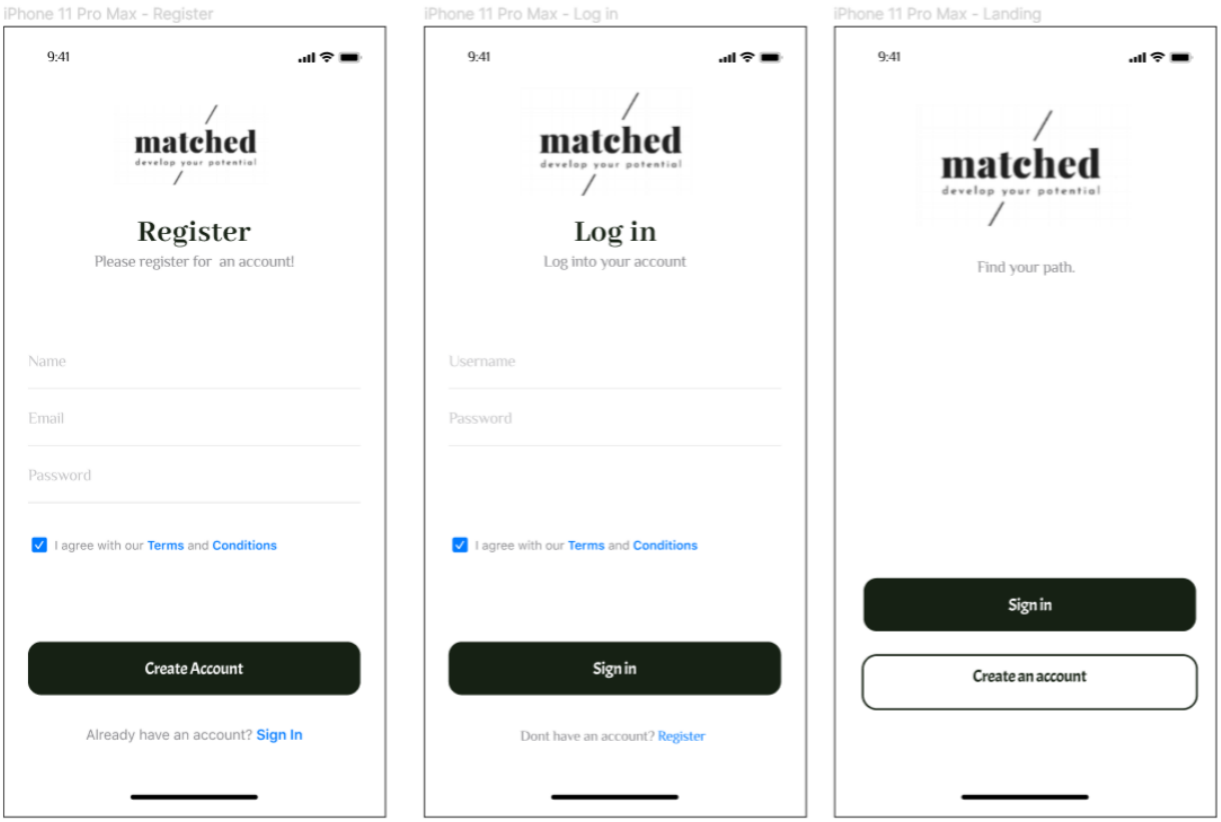


Fig1: The login and registration UI.

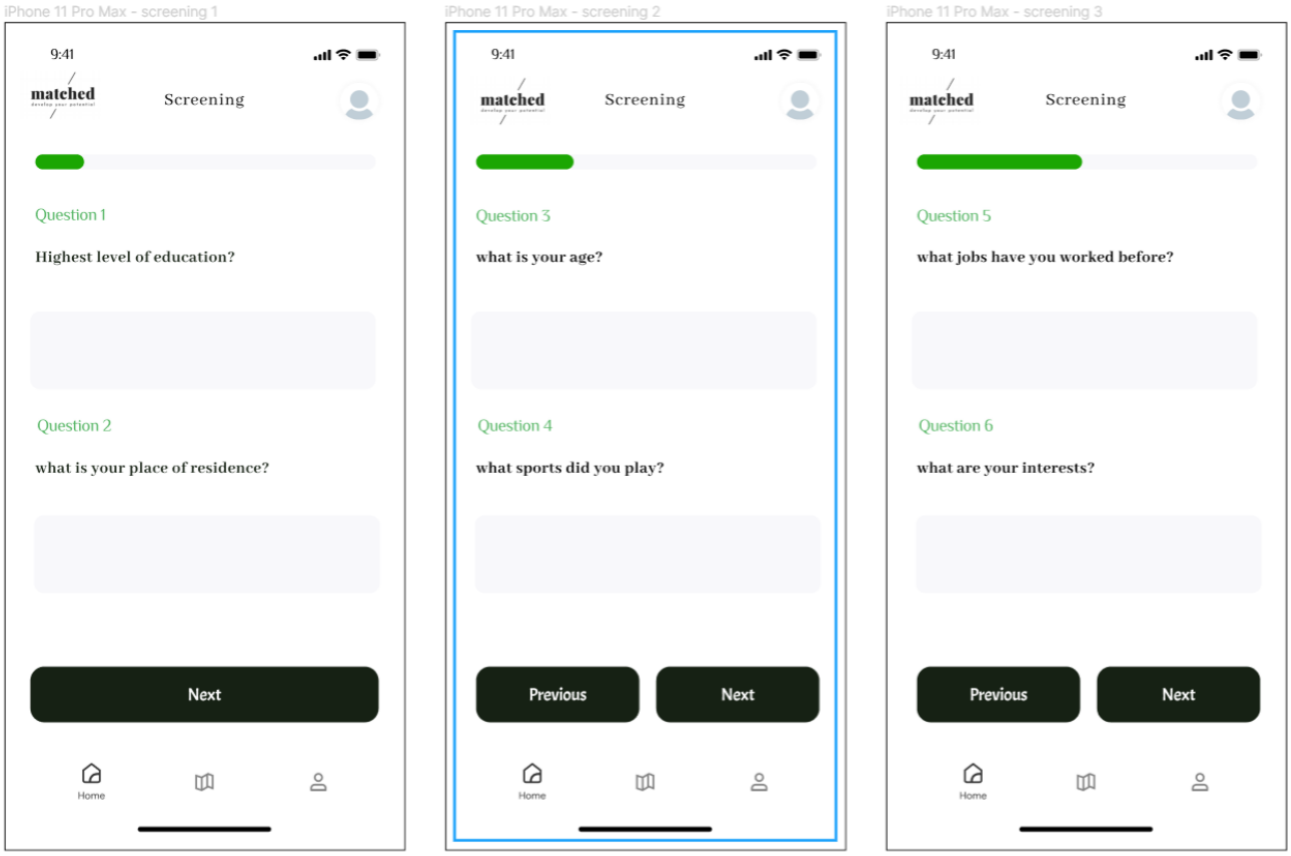


Fig2: Screening for the users.

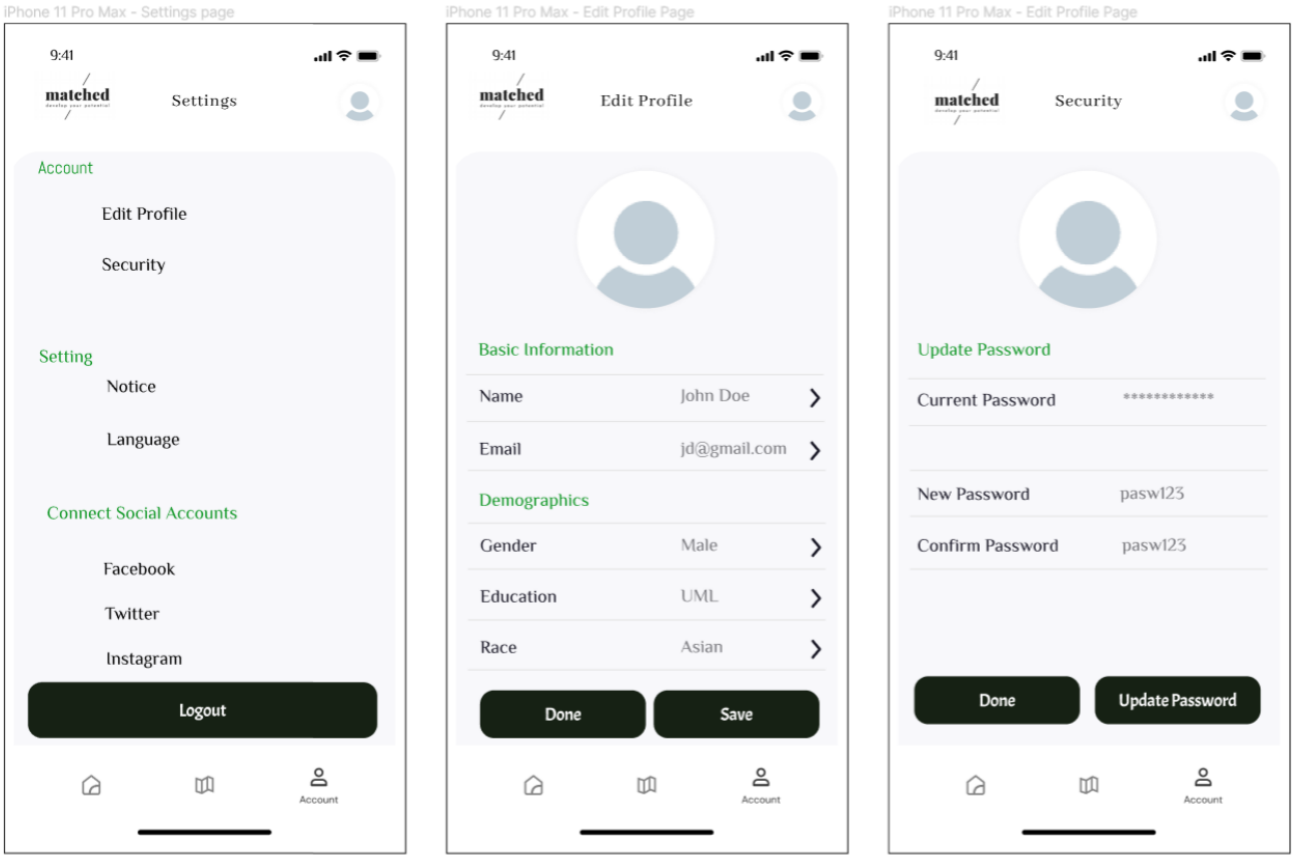


Fig3: Profile UI model of the users.

4 Time-frame and Milestones

1. **Project Proposal Revisions**
   - If needed- 2/25/22 to 3/25/2022
2. **Workers**
   - Will try to finish working on the registration page to grab the all the data of user and place in the right group in Django. - April 4th
   - Will try to finish the dashboard setup for the the user view- April 14th.
   - Design the profile view for the user April 20th
   - Design the Resources page for the user. -April 25th
3. **Testing**
   - Unit and client testing for react JS and Django - April 15th
   - Unit and client testing for react JS and Django - April 21th
   - Unit and client testing for react JS and Django - April 26th
4. **Deployment**
   - Unit and client testing for react JS and Django - May 4th
   - Final Deployment- May 5th.

5 Supplemental Documents and Notes

- Django Documentation: https://docs.djangoproject.com/en/4.0/

- React JS documentation: https://reactjs.org/docs/getting-started.html
- Python Documentation: https://docs.python.org/3.10/library/index.html
- CSS Documentation: https://www.w3schools.com/cssref/
- Bootstrap Documentation: https://getbootstrap.com/docs/5.1/getting-started/introduction/